Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 162 (2006) 255–259
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Pervasive Process Calculus

Robin Milner

The Computer Laboratory
University of Cambridge
Cambridge, UK

Abstract

Process calculi with various signatures and reaction rules may provide a theoretical basis for pervasive computing.

Keywords: bigraph, graph-rewriting, mobile agent, pervasive computing, process calculus.

Do process calculi need to ramify, if we want to model pervasive computing? I believe so; we can't expect a fixed small set of primitives to work for everything. This already shows up with locality and mobility (e.g. in the mobile ambients of Luca Cardelli and Andy Gordon) and with continuous dynamics (e.g. in the Φ-calculus of Bill Rounds). In general, it shows up when we try to decide which kinds of action are primitive; this depends on our purpose. Researchers who apply process calculi in biology are finding that the primitives they need depend upon the desired granularity of action; for example, at one level the ingestion of a molecule by a cell is a single discrete action, but at a lower level it is a continuous process. The same thing surely occurs in pervasive computing. For example, in a sentient building, the entry of a mobile agent into a room is at one level a single action, but at a lower level it is complex (involving keys, walking etc); and the computer in the room may then sense-and-login the agent in a single macro-action, involving detailed software activity at a lower level.

So we need a modelling framework which embraces calculi of processes at different levels; the realisation (or implementation) of a higher-level calculus by a lower-level one should be a kind of morphism of calculi. This is the aim of the *bigraphical* framework. I hope that these morphisms will harmonise with the concept of *refinement*, developed over many years by Tony Hoare and colleagues at Oxford. Then an important consequence could be a bigraphical language that coordinates descriptions on different levels – some for specification, some for programming. As well as this 'vertical' coordination, it should coordinate descriptions 'horizontally',

doi:10.1016/j.entcs.2005.12.112

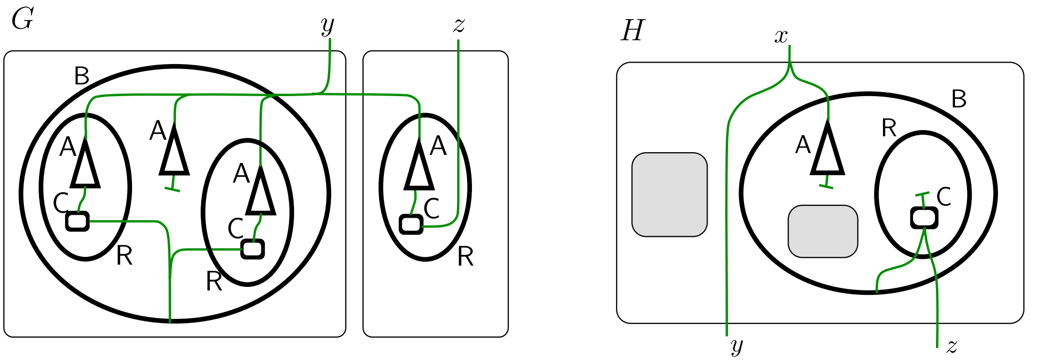

Fig. 1. Sentient buildings: structure. *The bigraph G is a subsystem of a larger system, and has two parts that may be widely separated in the whole. H is a host system; a context whose holes (the grey rectangles) may be occupied by the parts of G to form a larger system. The composite system $H \circ G$ is shown in Figure 2.*

in the familiar sense of coordination languages.

I am working on this with a team at the ITU Copenhagen led by Lars Birkedal and Thomas Hildebrandt, `http://www.itu.dk/research/theory/bpl`. We hope to derive a Bigraphical Programming Language (BPL) from the bigraph model. Then engineers of pervasive systems may find themselves programming in a language that is amenable to analysis because it is theoretically understood, though they need not know the theory. That was what ALGOL 60, logic programming and functional programming all aspired to; there is no reason not to aspire to it in the anarchical world of pervasive computing.

The bigraph model arose from the increasing need in real-world informatic systems, especially pervasive systems, to treat localities in the style of mobile ambients, and equally to treat the connectivities of a system in the style of the channels of CSP, CCS or π-calculus. Moreover (1) connectivities and localities may be real or virtual, (2) they should be treated independently of one another (*where you are does not affect whom you may talk to*), and (3) mobility is just reconfiguration of both these structures. What distinguishes bigraphs is not the details of these two structures, but the orthogonal treatment of them.

The rest of this note shows how bigraphs may capture the phenomena of pervasive computing, and also how they represent familiar process calculi rather directly.

First, consider a kind of sentient building B containing rooms R. Each room contains a computer C that is also a sensor, and all these computers are linked as part on the building's infrastructure. Agents A may occupy the building, inside or outside rooms; they carry devices — e.g phones — allowing them to communicate with each other; with the help of these devices they can also be sensed and logged in by any computer in a room. Figure 1 shows a bigraph G representing a system of such entities; the left part of G has a building with rooms and agents, while the right part has a single room, which may be somewhere else entirely, and a single agent. The nesting of nodes represents locality, and the slender links represent connectivity. One of G's links connects all the agents, even the one in the right part of G remote from the building; it may represent an ongoing conference call.

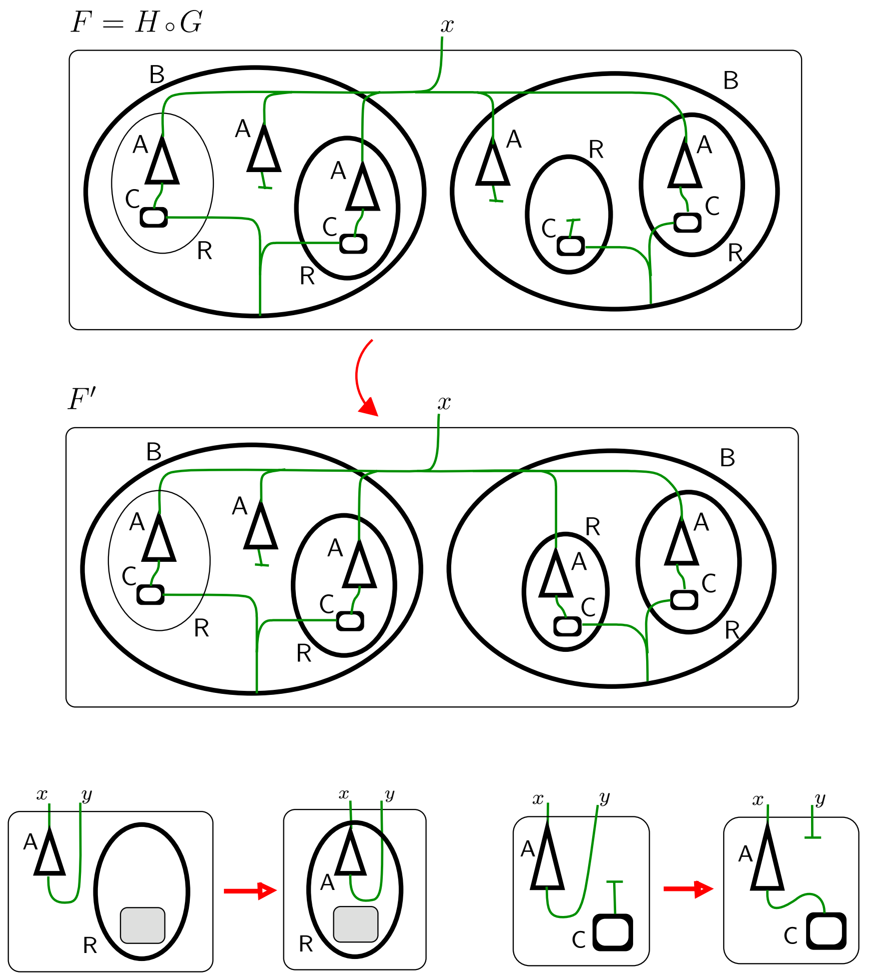

Fig. 2. Sentient buildings: dynamics. *$F = H \circ G$ represents the bigraph G of Figure 1 inserted in the host H. Note how the 'outer' names of G have been linked to the 'inner' names of H. The upper diagram shows a reaction in which F becomes F'; an agent* A *enters a room* R*, then it is sensed and logs in to the computer* C *located there. Below it are shown the two simple reaction rules that induce this reaction; one represents entry to a room, the other represents sensing and logging in.*

Another link connects the building's computers, as part of its infrastructure.

The number of G's regions, together with its 'outer names' y and z, constitute its outer (inter)face $J = \langle 2, \{y, z\}\rangle$. These interfaces are objects in a symmetric monoidal category, whose arrows are bigraphs. In general, an arrow is a *context*; it has not only an outer face, as G does, but also a non-trivial inner face determining the bigraphs with which it may be composed (i.e. those that can occupy the context). Thus $H : J \to K$ is such a context, where $K = \langle 1, \{x\}\rangle$. The inner face of G is the trivial one $I = \langle 0, \emptyset\rangle$; such a bigraph is called *ground*. To form the composition $F = H \circ G$, shown in the upper diagram of Figure 2, we insert the parts of G in the holes of H, and then join like-named links (one for y, one for z) and delete their names.

Dynamics is represented by a *reaction relation* over ground bigraphs, generated by a set of *reaction rules* which may vary from one application to another. Indeed, each application of bigraphs, called a *bigraphical reactive system* (BRS), is represented by a *signature* that specifies a set of *controls* (like B, R, $\ldots$) each having a few attributes, together with a set of reaction rules that these controls. Two very simple reaction rules for sentient buildings are shown in the lower paret of Figure 2; the first allows an agent to enter a room (preserving its linkage), while the second allows the room's computer to sense and login the agent. Note that a rule can be *parametric*; the grey holes in the first rule stand for arbitrary ground bigraphs with appropriate interfaces. Also note that the second rule requires the agent and computer to be co-located; it cannot apply between an agent in a corridor and a computer in a room.

From one viewpoint the actions represented by these rules are very elementary; at a higher level we may consider as primitive an action such as 'an agent A locates agent B in room R and moves there', which employs our rules (and others) many times. From another viewpoint the two rules are themselves complex; for example, entry to a room may involve a key and walking, while sense-and-login may invoke detailed software. Thus other BRSs may treat more complex or less complex activity as primitive. For example, if the building's operating system is written in a language based upon the π-calculus, then we would like to coordinate the building BRS with one that represents that calculus.

Joachim Parrow and others have exposed the topographical intuition underlying the π-calculus, using various graphical models. It is also easy to represent the π-calculus in bigraphs. In particular, parallel composition is represented by juxtaposition of nodes that may be linked (representing name-sharing). An idea of the encoding is given by Figure 3, which shows two reaction rules. The first represents the π-calculus rule

$$(\overline{x}z.P + \cdots) \,|\, (x(y).Q + \cdots) \longrightarrow P \,|\, Q \;;$$

two of its parameters (the holes) represent P and Q, while the other two represent the summands ('$\cdots$') that are discarded. (Bound names like y have a natural treatment in bigraphs; their scopes are locations.) Most of the axioms for structural congruence become just identities in bigraphs, which is an indication that bigraphs are faithful to the intuition of the calculus. However, replication —sometimes represented in the π-calculus by the axiom $!P \equiv P \,|\, !P$ of structural congruence— is best treated in bigraphs as a dynamic rule, as shown in Figure 3. Ole Jensen and I have worked out the correspondence between this encoding and the original calculus; the detailed story will appear in his forthcoming PhD dissertation.

This note only gives a rough idea of how bigraphs work. It shows that the model represents mobility of both placing and linking, that these places and links may be both physical and virtual, and that various different systems (with different signatures and reactions) can be coordinated in the bigraphical calculus. more detail can be found via my web page, `http://www.cl.cam.ac.uk/users/rm135`, which gives links to papers on bigraphs. Those written so far have concentrated upon recovering as much as possible of the theory of known calculi within bigraphs. Theory has been recovered for the π-calculus, mobile ambients, CCS, Petri nets

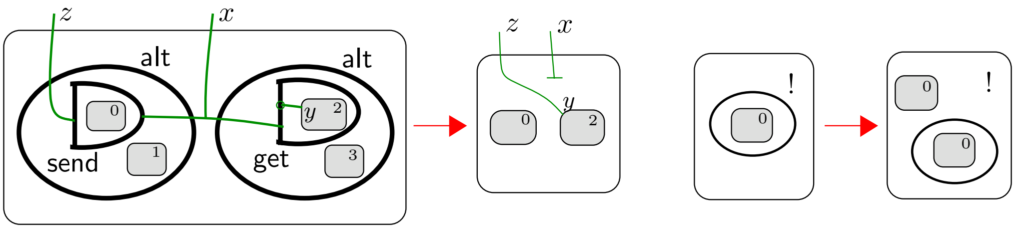

Fig. 3. Two reaction rules for the π-calculus. *The first represents communication; the controls* send, get *and* alt *represent sending, receiving and summation. The labelling of holes shows which parameters are discarded. In bigraphs replication is best presented as a dynamic rule, using the control '!'; here the labelling shows that the parameter is duplicated.*

and the λ-calculus; much of this work uniformly derives —from the reaction rules— labelled transition systems whose labels are small contexts. This derivation was worked out with Jamey Leifer, and exploits category theory in a novel way.

Two other approaches to this kind of work deserve mention. One is the long-existing theory of graph-rewriting based upon the double-pushout construction originated by Hartmut Ehrig and colleagues in the 1970s. That work is based upon categories with graphs as objects and embeddings as arrows. In bigraphs we have preferred graphs as arrows and interfaces as objects, since it closely follows the algebraic tradition of process calculi. But there are close links between the two approaches. Equally there are close links with the approach of Vladi Sassone and Pawel Sobocinski, based upon 2-categories. The approaches are complementary; work is in progress to see which approach works best for different purposes.

On the practical side we shall experiment with using bigraphs in applications, especially in the role of a programming language. If a language based upon such a topographical model is found convenient for specifying and programming pervasive computing systems, then it can contribute greatly to their scientific explanation. Given the complex nature of such systems, as well as the intimacy with which they will pervade our lives, such rigorous understanding is of the greatest importance.